Increasing Fairness and Capacity using MadMac Protocol in 802.11-based Ad Hoc Networkds

Tahiry Razafindralambo, Isabelle Guérin-Lassous

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Increasing Fairness and Capacity using MadMac Protocol in 802.11-based Ad Hoc Networks

Tahiry RAZAFINDRALAMBO — Isabelle GUÉRIN-LASSOUS

N° 5633 – version 2

version initiale July 2005 – version révisée October 2005

Thème COM

ISSN 0249-6399 ISRN INRIA/RR--5735--FR+ENG

Increasing Fairness and Capacity using MadMac Protocol in 802.11-based Ad Hoc Networks

Tahiry RAZAFINDRALAMBO*, Isabelle GUÉRIN-LASSOUS

Thème COM — Systèmes communicants
Projets ARES

Rapport de recherche n° 5633 – version 2† — version initiale July 2005 — version révisée October 2005 28 pages

Abstract: The IEEE 802.11 MAC layer is known for its unfairness behavior in *ad hoc* networks. But introducing fairness in the 802.11 MAC protocol may lead to a loss of the global throughput. Some solutions are proposed in the litterature to solve the fairness issues of 802.11 but these solutions only solve some specific problems and often lead to a poor global throughput. Our MadMac protocol deals with both global throughput and fairness and maximizes throughput when unfairness is solved. Fairness provided by MadMac is only based on information provided by the 802.11 MAC layer and its behavior is not probabilistic. MadMac has been tested in all configurations that are known to induce fairness issues. In most of these configurations, MadMac provides the maximum aggregated throughput that can be reached when a fairness scheme is adopted.

Key-words: fairness, capacity, *ad hoc* networks, 802.11, MAC protocols

* financed by France Telecom R&D
† Some figures of the previous version are corrected.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Augmentation de la capacité et de l'équité avec le protocole MadMac dans les réseaux *ad hoc* basés sur 802.11

Résumé : Le protocole MAC de 802.11 est connu pour son comportement inégalitaire dans les réseaux *ad hoc*. Réduire l'inégalité dans 802.11 provoque souvent une perte de débit global. Beaucoup de solutions sont proposées dans la littérature pour résoudre ce problème d'équité, mais ces solutions répondent souvent à des problèmes spécifiques et donnent souvent un faible débit global. Le protocole MadMac, que nous proposons, se veut équitable tout en s'approchant de la capacité du réseau. L'équité obtenue dans MadMac utilise seulement les informations fournies par 802.11 et son comportement n'est pas probabiliste. Nous avons testé MadMac sur toutes les configurations connues pour être inégalitaires, et dans la plupart des cas, MadMac permet d'obtenir le débit maximum agrégé quand l'inégalité est résolue dans le réseau.

Mots-clés : équité, capacité, réseaux *ad hoc*, 802.11, protocole MAC

1 Introduction

Ad hoc network have become more and more popular and many research problems, such as routing, quality of service, security, etc., are tackled. Most of the current ad hoc networks are based on the 802.11 standard [13] owing to the fact that this is the most widespread technology in the field of wireless local networks. Recently, different studies have shown some performance issues with the 802.11 Distributed Coordination Function (DCF) protocol, used for ad hoc network. These studies show that the origin of the performance problems comes from the MAC layer of 802.11. These performance problems are very often the consequence of unfairness and global performance loss [5].

Several solutions have been proposed to improve 802.11 performance by reducing unfairness issues or by improving global throughput in wireless ad hoc network. Recently, severals approaches try to increase both throughput and fairness by modifying the 802.11 MAC layer. Most of these solutions are based on rate and topology information exchanged between the nodes. The proposed protocols, not based on this kind of information, either reduce the fairness issues to the detriment of the aggregated throughput or increase the overall throughput without solving the fairness issues. In [20], the authors investigate the trade-off between aggregated throughput and fairness. They propose a model to compute the maximum aggregated throughput under various fairness schemes. This maximum global throughput, called fair capacity in our article, is the reference when evaluating protocols that deal with fairness issues. However, it is still a real challenge to design a fair MAC protocol for ad hoc networks that is distributed, topology independent, that relies on no explicit information exchanges and that present aggregated throughput close to this fair capacity.

In this paper we propose a solution to this challenge by designing a new protocol, called *MadMac*, that increases fairness in 802.11-based ad hoc network while maintaining an aggregate throughput in the network close to the fair capacity. Contrary to most of the previous solutions, our protocol is not based on a probabilistic waiting time. One of the main advantages of MadMac is that it is easy to implement because it is only based on information provided by the 802.11 MAC layer.

In Section 2, we present a state-of-the-art on the protocols that try to increase the capacity of ad hoc networks or/and that try to decrease the unfairness issues. In Section 3, we discuss the notion of fair capacity. The protocol MadMac is described in Section 4. Our protocol is then evaluated, in Section 4.3, in several well-known configurations that present fairness or performance issues. We show that our protocol has very good performances in all these topologies and solve many problems, like for instance the performance anomaly of 802.11 [3]. In Section 5, we discuss the impact of the fine tuning of different parameters of MadMac. Lastly, we conclude our paper with the outline of our future works.

2 Related Work

2.1 Capacity

Many papers study the capacity of wireless ad hoc networks that is considered as being the maximum global throughput that can be injected in the network. Some algorithms to achieve this bound are proposed. The first authors to study capacity in this context were Gupta and Kumar in [11]. In [18], the authors compute the maximal theoretical capacity offered by 802.11 in some specific scenarios. The authors of [2] design a distributed algorithm to compute a k-approximation of the network capacity.

Many others papers try to improve the capacity of 802.11 wireless ad hoc networks by modifying the 802.11 MAC layer protocol. The Binary Exponential Backoff (BEB) process used by the MAC 802.11 DCF is often modified to achieve better performance in terms of capacity. In the MACAW protocol [27], the BEB algorithm is compared to the MILD (Multiplicative Increase Linear Decrease) algorithm. The MIMLD algorithm [22] is the same as the MILD algorithm but with a multiplicative decrease phase as a first decrease step before the linear decrease phase. The two algorithms achieve better aggregated throughput than 802.11.

The authors of [17] introduce the notion of Distributed Contention Control (called DCC) to improve the performance of 802.11. With DCC, each station regularly computes a value called $slot\_utilization$, i.e the ratio between the number of busy slots over the number of available slots during a period of time. The $slot\_utilization$ associated to the number of unsuccessful previous transmissions is used to compute a probability of transmission. This mechanism significantly reduces the number of collisions and thus improves the performance of the 802.11. In [23], the authors give the optimal value of $slot\_utilization$ that represents the best compromise between the time spent in collision and the time spent in idle mode.

Other modifications have been proposed based on $p-persistent$ protocols [8]. A $p-persistent$ 802.11 protocol differs from the 802.11 protocol only in the selection of the backoff interval. At the beginning of an empty slot a station transmits its frame (in the current slot) with a probability p while the transmission is deferred with a probability $1-p$. The process is then repeated at the next empty slot. Basically a $p-persistent$ protocol is close to 802.11 MAC protocol from a capacity point of view. In [9] and [23], the authors present a dynamic $p-persistent$ protocol, called Simple Dynamic Protocol (SDP) to increase the 802.11 capacity and express the theoretical capacity limit of a $p-persistent$ 802.11 protocol. The protocols try to dynamically tune the backoff interval based on an estimation of the network status. This estimation requires the knowledge (or an estimation) of the number of active stations to reach an optimal performance. In [24], the authors compute the p value of a $p-persistent$ protocol to obtain p_{opt} that allows to reach the maximal capacity of the protocol. Authors also show that the product of the number of active stations M by p_{opt} leading to the optimal performance is asymptotically constant. In [16], the authors use this result to derive the optimal $slot\_utilization$ ([17]) value, called Asymptotic Contention Limit (ACL), which only depends on the average frame size and not on the number of active stations. In [16, 25], the same authors present two $p-persistent$ dynamic protocols, AOB

and AOB-enhanced that use the ACL function. AOB schedules its frame transmission like in classical 802.11 but adds a control level before the real frame transmission; the transmission is deferred according to a probability that depends on the channel utilization which optimal value is given in [9]. When a transmission is deferred AOB schedules the transmission following the 802.11 algorithm with a contention window multiplied by 2. Simulations on AOB show that capacity provided by AOB is greater than 802.11 capacity. AOB-enhanced is designed for networks running different MAC protocols (like 802.11 and AOB).

2.2 Fairness

Fairness issues on ad hoc networks have been deeply studied in the last years. Several mechanisms and protocols have been proposed to solve the fairness issues. There exist two main approaches in the literature. One approach is based on information exchanges between stations and/or a knowledge of the topology as in [6], [12], [21], [20], [26] and [15]. The other approach is topology independent and does not required any information exchanges ([4], [1], and [10]).

The authors of [21] present a mechanism for translating a given fairness model into its corresponding collision resolution backoff algorithm that probabilistically achieves the fairness objective but requires an efficient collision avoidance scheme (as RTS/CTS) to be efficient. Results show that on ring and clique topologies the proposed protocol achieves better fairness and more capacity than 802.11. In [20], the authors propose a packet scheduling scheme to achieve a fair and maximum allocation channel bandwidth. The algorithm proposed by the authors computes a scheduling based on a backoff modification. Their algorithm requires a knowledge of the topology and an exchange of flow information between nodes. In [26], a $p_{i,j} - persistent$ protocol where each station computes an access probability on the link between i and j is proposed. The backoff window size is computed according to information about their contention window size received from active neighbors. The authors of [12] try to enforce the max-min fairness by using an algorithm that compute the fair share. This algorithm requires the knowledge of the two-hop neighbors for each node to be efficient. In [6], the authors propose a backoff algorithm to improve both throughput and fairness. This algorithm requires the estimate of the number of active stations and a mechanism to avoid hidden terminal problem and is designed only for single hop networks. The EHATDMA protocol [15] is based on information exchanges initiated by the sender and/or the receiver before the data transmission to avoid the hidden terminal problem and leads to a better fairness than the protocol proposed in [26].

To cope with the lack of information on topology or from others nodes, some protocols base their decision on the data packets sent in the network only or introduce a probabilistic behavior in the nodes. In [1], each station adjusts its contention window size depending on its share of the medium with its neighbor nodes. This share is computed according to the number of sent packets by the station and the number of received packets from its neighbors. Results given in this paper and in [28] show that the algorithm proposed is better than 802.11 from the fairness point of view, but not for the aggregated throughput. The problem with this protocol is that the share of the radio medium for a station only considers the neighbor

nodes and not the nodes within the carrier sensing range. In [10] a distributed fair MAC protocol (FMAC) solves this carrier sensing problem. The main principle of FMAC is that the contention window size is tuned to reflect the number of successful transmissions during a time interval. Result given in this paper show that this protocol improves fairness but clearly reduces the network throughput. The authors of the PNAV protocol [4] introduce a fixed wait time between two successive transmissions depending on a probability. This probability depends on past events in the network. Results on PNAV shows that PNAV improves fairness on some topologies compared to 802.11, but PNAV throughput is always smaller than the 802.11 aggregated throughput.

2.3 Discussion

Our aim is to find the best trade-off between fairness and capacity. As far as we know, only one paper deals with the trade-off between these two notions, but the proposed algorithm requires a knowledge of the topology and an exchange of flow information between nodes [20]. We think that this approach is not the most efficient since information exchanges reduce the global throughput of the network. For example, a mechanism like RTS/CTS, that can be seen as an information exchange between nodes, decreases the global throughput of the network. We will show for instance that, with our proposed protocol, the RTS/CTS mechanism used to solve hidden terminal problem can be replaced by an appropriate fairness scheme. However, it appears from the literature that designing a MAC protocol that does not require any knowledge of the topology or specific information from other nodes than those provided by the MAC 802.11 protocol and the data traffic in the network is still a real challenge.

Most of the algorithms proposed to improve capacity and fairness depend on a random process. This probabilistic feature is effective either on the triggering of the modification or/and on the modification process or/and over the sending of packets. For instance, in the algorithm of [21], the triggering of the modification is random, the choice of the backoff is random and the sending of a packet is random since the protocol is $p - persistent$. This probability strongly depends on the network status. We have choosen a different approach since our algorithm tries to avoid the use of probabilities by introducing a deterministic behavior, in order to better control the protocol.

Finally, the literature shows that there exists a set of basic scenarios that lead to fairness issues with 802.11 in an ad hoc context [5]. Many of the previously quoted papers did not carry out their test on all these scenarios and many proposed solutions are specific to some configurations. One of our aim while designing our algorithm is to find a solution for fairness issues in many cases as possible.

3 Fair capacity

The trade-off between the aggregated throughput and fairness have been initially discussed in [20]. We think that the notion of capacity under a fairness scheme introduced in [20]

is fundamental to evaluate the performances of fair protocols that are proposed in ad hoc networks. We think also that this notion is not enough considered. That's why we discuss this trade-off in this section. We use a slightly different model than the one given in [20]. To simplify, we assume that the packets have the same size (like in [20]) and that the rates of the stations are equal. We consider the graph $G = (V, E)$ defined as follows:

- V, the set of nodes of G, represents the set of links of the network where a flow goes through: if there is a flow between the two neighbor mobiles i and j, then there is a node l_{ij} in the graph G.
- E, the set of edges of G, represents the set of interfering flows in the network: if the flow between the two neighbor mobiles i and j interfere with the flow between the two neighbor mobiles k and h, then there is an edge in G between the nodes l_{ij} and l_{kh}.

We consider that two flows interfere is they can not be emitted at the same time or if their simultaneous transmissions will collide on at least one receiver of one of the flows. These two conditions depend greatly on the underlying MAC protocol. Some CDMA based protocols may allow any node to transmit at any time, therefore two mobiles can always transmit at the same time. With CSMA/CA protocols, the first condition is the core of these techniques. With the MAC protocol of 802.11 that we consider, the existence of an edge between two nodes in the graph G means that either the two flows can not be simultaneously emitted or their simultaneous transmission may result in a corrupted transmission for at least one of the flow.

The capacity of an ad hoc network with initial traffic corresponds to the maximum throughput that can be achieved with this traffic injected in the network. Under the proposed model, the capacity is linked to the cardinality of a maximum independent set of the associated graph G and corresponds to the medium capacity multiplied by this cardinality. The radio medium capacity is considered as being the maximum throughput that can be achieved on this medium.

To illustrate this notion, let's consider Figure 1. There are two flows in the network and the emission of the flow between C and D will collide with the flow between A and B on B. We have thus in the associated graph G two nodes corresponding to the two flows that are connected since these two flows collide on B. Therefore the maximum independent sets are of size 1 and the capacity of this network with this traffic corresponds to the medium capacity.

Now, let's consider Figure 2. There are three flows in the network. The two flows l_{AB} and l_{DC} can be emitted at the same time whereas the flow l_{BC} interfere with the two other flows. We have thus the associated graph G given in the figure. Therefore the maximum independent set is of size two and the capacity of this network with this traffic corresponds to two times the medium capacity. It's interesting to note that, with this scenario, this notion of capacity is not fair. Indeed, the only way to reach the capacity is that the two exterior flows are always emitted and no packet is emitted on the central flow.

In [20], the authors introduce different schedules that are based on different fairness schemes while maximizing the aggregate throughput. Henceforth, we will call this concept

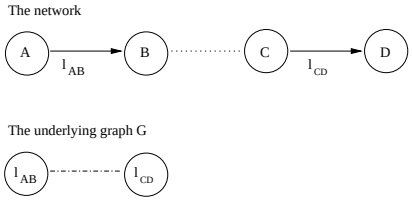

Figure 1: A first example and its associated graph G

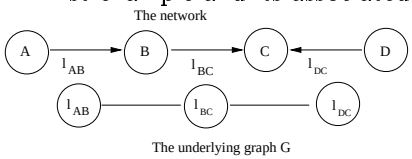

Figure 2: A second example and its associated graph G

the *fair capacity*. The fair capacity is the maximum global throughput that can be achieved in the network when a fairness scheme is adopted. This concept should help to measure the performance of fair protocols for ad hoc networks by comparing the fair capacity and the overall throughput offered by the protocol. In many cases, the capacity of the network will be greater than the fair capacity. The authors of [20] design a schedule to achieve this fair capacity in the case of a local fairness model. The idea is to find flows that need to be scheduled and that are independent and then to maximize the aggregated throughput by maximizing the size of the independent set. After this schedule, only the flows that needed to be scheduled are marked. Thus, a local fairness is guaranteed while maximizing the overall throughput. Note that with our assumptions (the same packet size and the same rate), this scheduling is equivalent to coloring the associated graph G with the minimum number of colors and then maximizing the independent sets computed with the coloring.

Like many articles that deal with fairness in ad hoc networks, we have considered the max-min fairness scheme, as it is considered as the fairer scheme[1]. Most of the tested scenarios that present fairness issues are simple and it is very easy to compute their fair capacity. With these configurations, we always compare the achieved aggregated throughput with the fair capacity under the max-min fairness scheme rather than with the capacity since this notion is much more adapted to the protocols that try to increase the fairness among the flows.

[1] but not as the most efficient in terms of global performance. The discussions on the quality of the max-min fairness in the ad hoc context are out of the scope of this article.

4 MadMac: A Fair and Efficient Protocol

The approach of MadMac is to provide a schedule on the packets like the one designed in [20] but topology independent and with no extra information than the one provided by 802.11. Of course, a perfect schedule is difficult to obtain with these constraints but the simulation results will show that we obtain good performances.

4.1 Description

4.1.1 The basic scheme

The idea behind the proposed protocol comes from the following remarks:

- If an active node senses activity on the channel, then it means that it is not alone on the channel and that at least two stations (including itself) send packets on the radio medium.
- If an active node experiments one or more collisions on its packets, then we can derive the same conclusion: at least two stations (including itself) send packets on the radio medium.

The second statement differs from the first one in the sense that the detected competing stations are not necessarily in communication or in carrier sensing range. However, we can say that, from the point of view of the node that experiments collisions and as stated in Section 3, they share the medium since the station can not successfully send its packets due to interfering transmissions. Note that, considering only the sensing activity and/or the experimented collisions, a node can not determine how many nodes compete with it. To approximate this number, other operations are required like capturing useful data (the source and the destination for instance) in control and data packets. However it seems difficult to exactly deduce this number as soon as a carrier sensing mechanism is used. Since we don't want to use and send extra information, each node can only deduce, with these two statements, whether it shares the medium (in the general sense) with at least one another node.

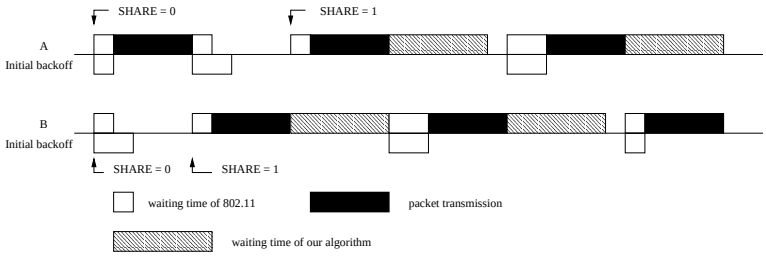

Figure 3: A simple illustration of our algorithm

If at least one of these statements is true, then the active node sets a boolean variable, called $SHARE$ to 1. Since the share is not permanent, this variable is updated periodically. We consider a period of $Delta\_Slot$ which the value will be discussed later on. At the beginning of each $Delta\_Slot$, the $SHARE$ variable is reset to 0. When $SHARE$ is equal to 1 for one node, then this node considers that it shares the medium with one or more stations and decides to reduce its MAC throughput by 2. The throughput division is done by introducing a waiting time before each packet to send. The goal of this waiting time is to introduce an alternate schedule between the competing nodes, and more generally to achieve such a schedule between the independent sets of the network as described in Section 3. At this point comes our first assumption: as no information is extracted from the packets, each node assumes that the competing mobiles have the same packets size as its own packets. This assumption will reduce the overall throughput but not the fairness between the nodes as we will see in the following. With this assumption, the waiting time T_{WAIT} is equal to $T_{DIFS} + M + T_p + T_{SIFS} + T_{ACK}$, where T_p is the packet time transmission of this node, T_{ACK} is the ACK time transmission, T_{SIFS} and T_{DIFS} are respectively SIFS and DIFS duration and M is the mean backoff time of 802.11 (*i.e.* $310\mu s$). This waiting time is never stopped and is active for each packet that is not entered in the medium access process of 802.11 as soon as $SHARE$ becomes equal to 1.

After this deferring time, our algorithm uses the classical medium access algorithm of 802.11 for the packets to send, *i.e.* a waiting time of $DIFS$ plus a backoff time that can be stopped as soon as activity is detected on the radio medium and the binary exponential backoff algorithm. Note that a random access can not be removed from our algorithm for different reasons. First the considered statements only indicate that the medium is shared but not provide the number of competing nodes. Therefore the added waiting time can not perfectly shift the node with the competing stations. Moreover, the waiting time is based on a mean backoff and on an assumption on the packets size of the competing nodes. Thus, even if each node shares the medium with only one another station, the alternation of the emissions introduced by the waiting time is not perfect and sometimes the two stations will enter at almost the same moment in the medium access process of 802.11. Therefore, we have to keep a random access in the algorithm. However, since this extra waiting time should reduce collisions, we use a smaller contention window size than the CW_{min} of 802.11 (10 slots in our experiments).

Figure 3 gives a simple illustration of our algorithm. We assume that two nodes A and B are in communication range of each other. In this figure, the $Delta\_Slot$ value is greater than the time represented. We assume that the two associated receivers are also in the same communication range. The figure shows the process of each node. At the beginning, no station has sent a packet, then the $SHARE$ variable of the two nodes is equal to 0. Each node wishes to send a packet and then enters in the medium access algorithm of 802.11: it corresponds to the white box. We have simplified the figure by neglecting the $DIFS$ and only show the backoff time. The white boxes under the lines correspond to the initial backoffs drawn by the stations. Station A wins the contention and sends its packet (black box). We have also simplified the data exchange since $SIFS$ and the acknowledgment do

not appear on the figure. Station B detects the activity of station A and thus sets $SHARE$ to 1. However, its first packet has already entered in the 802.11's process, therefore it keeps on decreasing its backoff when the medium is idle without extra waiting time. Station B wins the contention and sends its packet. Then A sets its variable $SHARE$ to 1, but since its second packet has been entered in 802.11's process before this setting, 802.11's algorithm is still applied on this second packet of A. B after its first sent packet has its $SHARE$ variable set to 1. Therefore it adds a waiting time before the sending of its second packet (dashed box). At the end of this waiting time, B enters in 802.11's process and thus draws a backoff. Since the medium is free, B can send its packet after its backoff reaches 0. Meanwhile A has sent its second packet and then, since $SHARE$ is equal to 1, adds a waiting time before the sending of its third packet. This process is repeated so on for each node. We see that such a scheme allows to alternate the sending of the two competing nodes' packets.

4.1.2 Collision avoidance

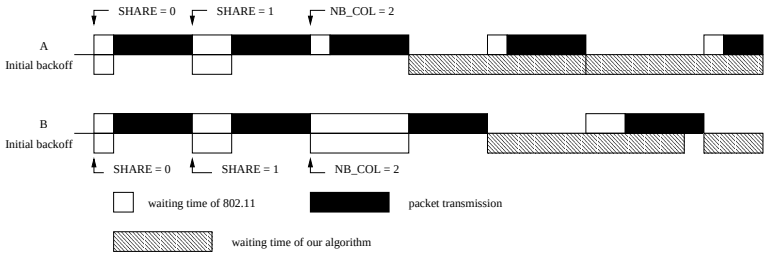

Figure 4: A simple illustration of our algorithm on the hidden terminal configuration

To manage collisions, we use the Binary Exponential Backoff algorithm of 802.11, but we keep track of the successive collisions: We use another variable called NB_COL. NB_COL maintains the number of successive collisions encountered by the node on its last transmission. This value is updated at every successful transmission and when an activity is sensed on the medium. If the node does not sense any activity on the medium, the value of NB_COL is set to 0 after the transmission. If this variable reaches a certain value k (a parameter of our algorithm), then we consider that the node that experiments such multiple collisions is very likely in a hidden terminal configuration. As shown in [19], in such a configuration, first each emitter has to send many packets before successfully sending the first packet, and then when one of the node wins the contention, it can send several consecutive packets. To avoid the overall throughput decrease due to many collisions and the short time unfairness due to the sending of consecutive packets, we force the two hidden nodes to emit in turn. For that, as soon as the node succeeds in transmitting the packet that has experimented at least k collisions, then we introduce another waiting time of at most $2 * T_{WAIT}$ for the following packets. This waiting time, called T_{ALT} henceforth, differs from T_{WAIT} in the sense that it is stopped as soon as the node senses activity on the medium (ACK for the

other node). If no signal is detected, then the node waits for T_{ALT}. After this step, when the medium becomes idle or when T_{ALT} is elapsed, the node enters in 802.11's medium access process (with a smaller contention window than 802.11) to send its packet. Thus, the nodes in competition will alternate their emission. This process is maintained while the node experiments no collision and detects activity on the medium during its waiting time. If no activity is detected, then the basic process is restarted.

Figure 4 illustrates the run of our algorithm on a hidden terminal configuration. We consider in this example that k is equal to 2. Node A and node B are the two hidden nodes that wish to communicate with the same receiver. At the beginning the variable $SHARE$ is set to 0 on the two transmitters. Then, we assume that they draw the same backoff, therefore there is a collision on the two sent packets. The two emitters double their contention window size since they follow the MAC protocol of 802.11 at this step. We assume that they also experiment collisions on this second emission of the first packet. The nodes still double their contention window size. We assume that A draws a shorter backoff and then wins the contention and succeeds in sending its whole packet. During the sending of A's packet, B decrements its backoff since they are hidden. When B sends its packet, it experiments no collision since the sending of A is ended. For the following packets, since $NB\_COL \geq k$ and the nodes A and B have respectively sensed the ACK of the parallel transmission, the two nodes start waiting for activity on the channel during a time interval of at most T_{ALT} (represented by the dashed boxes on the figure). During this period, A detects activity corresponding to the receiver's acknowledgment to the the first packet of B. A then stop its waiting and enters in 802.11's process to send its second packet. Since B is blocked by its waiting time which is large enough, A sends successfully its data. Therefore B senses the receiver's acknowledgment to A's packet and can then stops waiting and enters in 802.11's process to send its second packet. This pattern is repeated, and we can thus avoid collisions in such a configuration.

4.1.3 No monopoly on the channel

In some configurations, shown in [5], some nodes may monopolize the radio medium preventing some other stations from accessing to the channel. These nodes never experiment collisions and always sense the medium free since the other competing nodes don't succeed in accessing the medium. Our protocol, as we have just described it, does not provide solution for this kind of situations. Therefore, to avoid such a monopoly on the channel, we change for some packets the contention window size of these nodes. After every x consecutive successful packets sending with no deferring (*i.e.* $SHARE$ is always equal to 0), the contention window for the $x+1$th packet is set to 2 times the CW_{min} of 802.11 (*i.e.* 64), and to 4 times the CW_{min} of 802.11 (*i.e.* 128) for the $2x+1$th transmissions (x is a parameter of the protocol). This pattern is repeated for the following packets. This process should allow other nodes to access the medium and to send a packet, which will update the $SHARE$ variable of the monopolizing node.

4.2 Summary and discussion

Algorithm 1 gives the sketch of our MadMac protocol. During each *Delta_Slot*, each node either senses the channel, or receives a packet, or wishes to send a packet. If the node senses the medium busy or receives a packet then it sets its variable $SHARE$ to 1. This variable will be only updated at the next *Delta_Slot*. If the node has a packet to send, the protocol has different parts depending on what have previously happened.

- If no activity has been detected, then our protocol uses the MAC protocol of 802.11, denoted by the function *802.11-sending* in Algorithm 1. Note that this function may change the state of the variables $SHARE$ and NB_COL, since during the backoff decrementing the medium can be sensed busy or the packet can experiment one or more collisions. Anyway, whatever it may happen, it's always the MAC protocol of 802.11 that is used to send this packet.
- If activity has been detected and there has been none or few collisions on the previous sent packet, then the basic scheme of our protocol is used, denoted by the function *basic-sending*. Note that only NB_COL can be modified since $SHARE$ is already set to one and can only be updated in the following slot. In this part, the node mandatory waits during T_{WAIT} and then uses the MAC protocol of 802.11 to send its packet.
- If there has been more than $(k-1)$ collisions on the previous sent packet (note that NB_COL is updated if a packet is sensed on the medium), then we consider that this node is very likely in a hidden terminal configuration. Then we force the alternation of emissions with the function *hidden-sending*. In this part, the node starts waiting for a period of at most $T_{ALT} = 2 * T_{WAIT}$. If it detects activity during this period, then it stops waiting and uses the MAC protocol of 802.11 to send its packet. This scheme is repeated while the node detects activity.

In the following, we discuss the features of MadMac and the expected behavior and performance. First we can notice that our protocol is based on very simple operations (activity sensing and number of collisions) that is very easy to obtain with 802.11. No extra information are needed, like the number of neighbors for each node for instance.

The time on each node is divided into time slots of size *Delta_Slot*. Note that no synchronization is required on the time slots of all the nodes and the division is peculiar to each node (although the time slot size is the same on all nodes). Of course, the smaller *Delta_Slot*, the reactive the protocol. On the other hand, too small values of *Delta_Slot* will bring to the nodes a too short knowledge on the past activities. There is a trade-off to find.

When a node senses the medium busy or experiments a collision, then it assumes that the competing node or the competing nodes (it can not decide) have the same packet size. Some simulations have been carried out using different packet sizes. The results show that this difference does not degrade the fairness nor the throughput provided by MadMac because the introduced waiting time is large enough to fully decrement a backoff time. The k parameter is also important because collision may occur due to wireless channel variations and thus too

small values of k make the node enter the *hidden-sending* phase without being in a hidden terminal configuration. In MadMac this parameter is set to five because encountering five successive collisions is enough to consider that it is not due the the wireless channel state. Having k greater than five makes that the nodes never enter the *hidden-sending* phase because the backoff windows are large enough to allow a successful transmission in the backoff interval before experimenting six collisions.

```
NB_COL := 0;
for Each Delta_Slot do
    SHARE := 0;
    repeat
        if (SHARE == 0) and (medium busy or packet received) then
            Then SHARE := 1;
        if (packet to send) and (SHARE == 0) and (NB_COL < k) then
            Then 802.11-sending(SHARE,NB_COL);
        if (packet to send) and (SHARE == 1) and (NB_COL < k) then
            Then basic-sending(NB_COL);
        if (packet to send) and (NB_COL ≥ k) then
            while (activity detected) do
                hidden-sending(NB_COL);
    until;
```

Algorithm 1: MadMac protocol

4.3 Simulation results

The proposed protocol has been evaluated by simulations using NS-2 [14]. The comparison has been performed using 802.11. We have tested most of the scenarios presented in [5]. These studies have been carried out using a constant bit rate application that saturates the medium and a packet size of 1000 kbytes. We have modified some of the NS-2 parameters such as the power and the transmission range to reflect the HR-DSSS 11 Mb/s physical layer of the 802.11b protocol. To avoid message transmission other than those created by the constant bit rate traffic, a static routing agent is used. Other sources of traffic such as those generated by the ARP protocol have also been disabled.

4.3.1 Performance of one-hop networks

The first simulations have been performed on the simple scenarios where communications take place between nodes that are in communication range of each other.

In these scenarios there is no fairness issue. The goal of this section is to compare the performances of our protocol MadMac with 802.11 in this classical configuration. The results given on Figure 5 show that our protocol provides a higher overall throughput than 802.11.

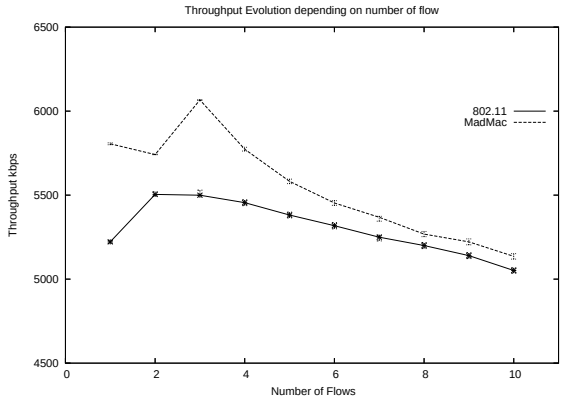

Figure 5: Total throughput evolution depending on the number of active nodes in an ad hoc cell

This is due to the fact that the contention window size is set to a lower value than in 802.11. In case of one single active node in the ad hoc cell, the contention window size is set to 64 for the $(x+1)th$ packet and to 128 for the $(2x+1)th$ packet (see Section 4.1.3). This extra time to avoid a monopoly of the medium is masked by the smaller size of the contention window than in 802.11 for most of the sent packets. In the simulation the x parameter (number of packets successively sent with a small backoff window) is set to 10.

The achieved global throughput with two active nodes is also higher than with 802.11, but is smaller than with one or three active nodes. This is due to the fact that the two nodes alternate their emissions due to the extra waiting time (T_{WAIT}) and that this alternation is almost perfect. Therefore the overlapping of the backoff decrementing is rare in this configuration. For scenarios with more than two stations, the last node that has sent a packet on the medium is the only one to enter in the waiting phase while the other nodes finish their waiting phase or enter in the 802.11's process, *i.e.* the backoff decrementing process (after a $DIFS$). Therefore, there is an overlapping of the backoff decrementing phases which leads to a smaller time interval between two consecutive packets sent on the medium than with two nodes.

We see also that the overall throughput of MadMac decreases with the number of contending nodes (like for 802.11), but is always higher than 802.11. This decreasing is due to the increase of collisions for the two protocols. As the contention window size of MadMac is smaller than the one of 802.11, the number of collisions with MadMac is a little bit higher. But we see that it does not drastically reduce the throughput and MadMac still presents good performance for ten nodes.

Henceforth, we consider that the radio medium capacity, obtained with MadMac, is the throughput achieved with one emitter and corresponds to 5.800 Mb/s. We will use this value in the following to derive the fair capacity of the tested scenarios.

4.3.2 The hidden terminal configuration

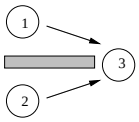

Figure 6: The hidden terminal scenario

One scenario involving a fairness issue is the well-known hidden terminal problem [19] depicted in Figure 6. In this scenario, node 3 is in the transmission range of both nodes 1 and 2 that are fully independent. The main problem with 802.11 on this scenario is that the number of collisions is high, which leads to an increase of the contention window size that drastically reduces the throughput of nodes 1 and 2. When the difference between the backoff of the two emitters is large enough to successfully send a frame for one node, then the contention window of the winner is set to the minimum. The difference between the two contention window sizes induces short time unfairness. Some mechanisms such as RTS/CTS have been proposed in the literature to reduce this problem but none of them makes it possible to obtain the maximum capacity that can be provided.

Figure 7 (Fig. 8 resp.) gives the results of hidden terminal simulations running through 40 seconds without (with resp.) RTS/CTS. First, these results show that our protocol removes the short time unfairness. This short time unfairness appears in Fig. 7 and 8 as the throughput variation seen with the 802.11 protocol. Second, our protocol provides greater by-node throughput and a greater total throughput (see also Table 1). This is due to the fact that, with MadMac, the hidden nodes almost perfectly alternate their emission, which does not result in many collisions. Therefore their contention window size remains low and the difference between these two sizes is also low. We can also see in Table 1 that the overall throughput achieved in this scenario is close to the one achieved in the one hop scenario with 2 active node (Fig. 5). We can notice that the short time unfairness is not solved with the RTS/CTS mechanism (see Figure 8) even if it reduces the number of collisions.

The fair capacity of this configuration corresponds to the medium capacity since the two flows are dependent in the associated graph G defined in Section 3 and that a max-min fairness scheme allocates a rate of $\frac{C}{2}$ (C corresponds to the medium capacity) to each flow. We see that MadMac offers an overall throughput very close to this fair capacity.

4.3.3 Another impact of the hidden terminal configuration

In the third scenario we propose to study another impact of the hidden terminal scenario depicted in Figure 9. This configuration has first been pointed out in [1]. In this scenario node 1 sends packets to node 2 and node 3 to node 4. Nodes 1 and 3 are hidden from each other. Nodes 2 and 3 are in communication range. In this topology, node 1 always senses the medium free and node 3 can transmit its packets when node 2 is not sending an ACK. On one hand when node 3 sends a packet, the transmission is always successful because no

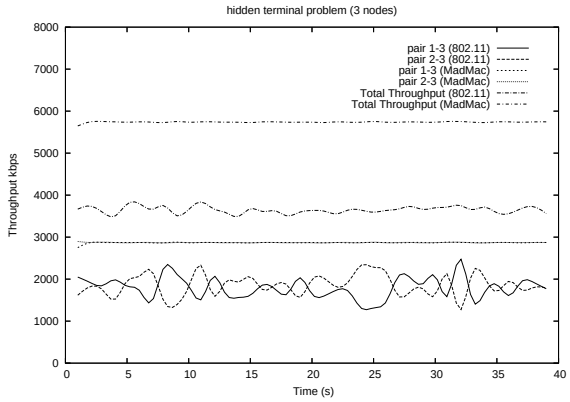

Figure 7: Hidden terminal scenario: individual and global throughputs without RTS/CTS

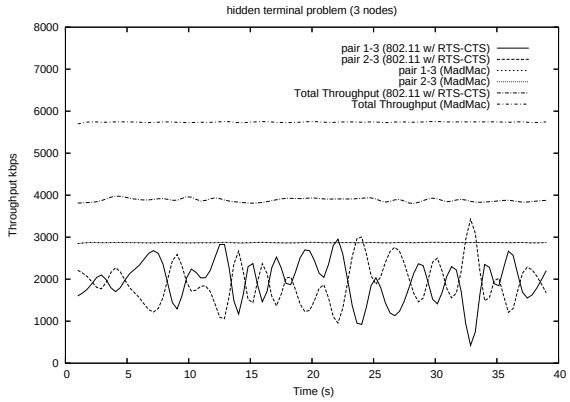

Figure 8: Hidden terminal scenario: individual and global throughputs with RTS/CTS

collision happens at node 4. On the other hand, when node 1 transmits its packet, there is a high probability that node 3 also transmits one, thus a collision will occur on node 2. The only chance for node 1 to successfully transmit a packet is when its frame is sent during a silent period of node 3. This successful transmission strongly depends on the frame length of node 1. A solution to this problem is to use the RTS/CTS mechanism that can reduce the number of collisions in 2 because the length of the RTS frames are often smaller than the data frames, but this use is not very efficient.

Results of the simulations are given in Table 2 and in Figures 10 and 11. First, let's consider that the RTS/CTS mechanism is disable. We can see from Fig. 10 and Table 2 that our protocol provides a global throughput smaller than the one offered with 802.11. Nevertheless, our protocol provides a better fairness since the throughput achieved by the

		Throughput (kbps)	Confidence Interval (0.05)
802.11	1-3	1844.81	[1776.42 - 1913.20]
	2-3	1782.98	[1717.46 - 1848.50]
	Total	3627.80	[3599.60 - 3655.99]
802.11 RTS/CTS	1-3	1961.62	[1815.33 - 2107.91]
	2-3	1921.05	[1776.52 - 2065.59]
	Total	3882.68	[3870.83 ; 3894.53]
MadMac	1-3	2867.09	[2861.77 - 2872.41]
	2-3	2871.06	[2869.34 - 2872.77]
	Total	5738.15	[5733.70 - 5742.61]

Table 1: Results on hidden terminal scenario

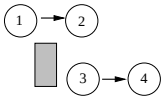

Figure 9: Another impact of the hidden terminal scenario

pair $1-2$ with our protocol is greater than the one achieved by 802.11. This is due to the waiting time introduced on the second emitter and that allows more successful emissions on the first flow. However, we have to note that the overall throughput of our protocol is smaller than the fair capacity. This configuration is identical as the hidden node configuration in terms of the flows dependency since the two flows are linked in the graph G and the fair capacity is equal to the medium capacity. This difference is due to the fact that collisions still exist on node 2 since the alternation is not perfect between the two emitters and since every $Delta\_Slot$ the two sources reset their $SHARE$ variable which leads to a direct emission of the packets without extra waiting time. Note that MadMac does not considered this configuration as a hidden node scenario since node 1 never detects activity on the medium even if it experiments collisions. A call to the collision avoidance phase of MadMac will not help much in this case since the alternation can not be triggered by the second flow that is completely masked to node 1.

The results of Figure 11 show that our protocol provides a better fairness and a greater overall throughput than 802.11 with the RTS/CTS mechanism. As with MadMac, using RTS/CTS does not avoid all the collisions. The throughput is lower than with MadMac due to the overhead induced by the RTS/CTS mechanism. These simulations (including the ones carried out on the hidden terminal scenario) show that it is possible to replace the RTS/CTS mechanism by an appropriate MAC scheme that leads to a better fairness and a higher global throughput.

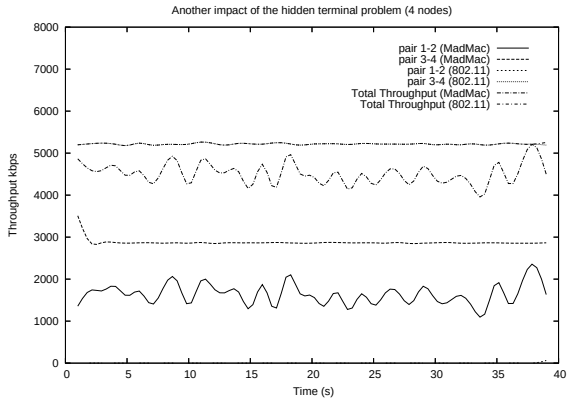

Figure 10: Another impact of the hidden terminal: individual and global throughputs without RTS/CTS

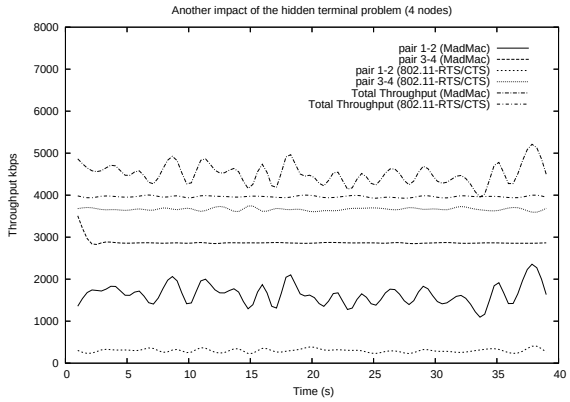

Figure 11: Another impact of the hidden terminal: individual and global throughputs with RTS/CTS

Lastly, it's interesting to note that MadMac increases the long term fairness but in the same time introduces a little short term unfairness. This is due to the fact that our protocol can not avoid all the collisions on node 2, which reduces its rate after these collisions.

4.3.4 The three pairs

The fourth studied scenario is the three pairs scenario depicted in Figure 12 and pointed out in [7]. In this scenario, the communications take place between nodes $1-2$, $3-4$, and $5-6$. Nodes 1 and 5 are fully independent and node 3 is in the carrier sensing range of nodes 1 and 5. With 802.11, it is easy to see that the backoff decrementing of node 3 can only take

		Throughput (kbps)	Confidence Interval (0.05)
802.11	1-2	0.0	[0.0 - 0.0]
	3-4	5215.70	[5210.91 - 5220.48]
	Total	5217.31	[5212.41 - 5222.21]
802.11 RTS/CTS	1-2	298.42	[286.34 - 310.49]
	3-4	3666.14	[3656.61 - 3675.66]
	Total	3964.56	[3959.01 - 3970.10]
MadMac	1-2	1634.75	[1564.54 - 1704.97]
	3-4	2879.22	[2851.26 - 2907.18]
	Total	4513.98	[4443.18 - 4584.78]

Table 2: Another impact of the hidden terminal scenario: results

place when nodes 1 and 5 are not transmitting and are in their silence period. Moreover the overlapping of these silence periods has to be greater than an EIFS (Extended Inter-Frame Space) since the central emitter is in carrier sensing range of the two external emitters.

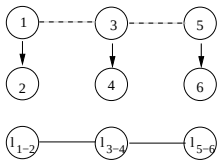

Figure 12: The 3 pairs scenario and the underlying graph

Simulation results on this scenario are given on Table 3 and Figure 13. In this figure, the rate of flow $5-6$ is not represented because of its similarity with flow $1-2$. We first see that MadMac is fairer compared to 802.11. We see also that the overall throughput is lower than the one of 802.11. We have here a typical example of trade-off between capacity and fairness, compared to the previous configurations where the capacity is equal to the fair capacity. Figure 12 gives the associated graph G on the flows dependence. In this case, the fair capacity is $3/2$ times the medium capacity (in a max-min fairness scheme, each flow has a rate of $\frac{C}{2}$), which corresponds to 8.7 Mb/s. This means that in this situation it is impossible to achieve both network capacity and fairness. We see that MadMac is very close to the fair capacity in this scenario, while 802.11 is close to the capacity of this network.

4.3.5 The chain configuration

This scenario is the chain topology depicted on Figure 14. A flow is sent from one extreme point of the chain to the other extreme point. Note that it is different from the scenario where an independent flow runs on each link of the chain. This scenario have been studied in [18]. In this paper the authors say that the maximum end-to-end capacity that can be

		Throughput (kbps)	Confidence Interval (0.05)
802.11	1-2	5095.39	[5074.84 - 5115.93]
	3-4	135.37	[117.09 - 153.64]
	5-6	5100.41	[5081.49 - 5119.33]
	Total	10331.18	[10309.71-10352.66]
MadMac	1-2	2863.53	[2862.36 - 2864.70]
	3-4	2863.53	[2862.60 - 2864.46]
	5-6	2863.53	[2862.36 - 1864.70]
	Total	8590.59	[8588.00 - 8593.19]

Table 3: 3 pairs scenario: Results

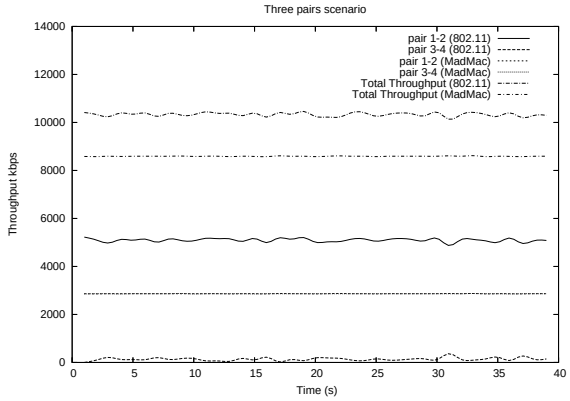

Figure 13: The 3 pairs scenario: individual and global throughputs

achieved by a perfect MAC protocol (based on CSMA/CA) on chains of at least 4 hops is 1/3 times the medium capacity if the communication range is equal to the carrier sensing (case 1) range and 1/4 times the medium capacity if the communication range is two times smaller than the carrier sensing range (case 2).

Simulation results on this scenario are given in Table 4. Chain lengths from one hop to seven hops are evaluated. We see that MadMac gives a better throughput, except for a chain of four hops.

The conclusion that we can bring out from these simulations is that it is very difficult to study the chain topology due to the different mechanisms that are involved in this scenario. As an example we can see that with 4 hops, 802.11 can achieve more than the theoretical capacity defined in [18]. This is due to the fact that link $\{l_{1,2} - l_{4,5}\}$ can be triggered nearly at the same time and the transmission can be successful in NS2 if $l_{1,2}$ is triggered a little bit earlier than $l_{4,5}$. Reception will be successful at node 2 because the SNR (Signal to Noise Ratio) is large enough to correctly decode the frame. With MadMac, due to the extra waiting time, the probability that $l_{1,2}$ is triggered before $l_{4,5}$ is very low. Results on this scenario when running case 1 are closely the same as in case 2. Increasing the MadMac throughput to obtain the theoretical capacity in this scenario is clearly our next (but not so easy) challenge.

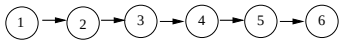

Figure 14: Chain scenario

		Throughput (kbps)	Confidence Interval (0.05)
802.11	1-2	5221.77	[5214.04 - 5229.50]
	1-3	2567.47	[2564.02 - 2570.91]
	1-4	1856.29	[1848.62 - 1863.96]
	1-5	1401.84	[1394.17 - 1409.52]
	1-6	1175.68	[1158.36 - 1192.99]
	1-7	1099.52	[1089.49 - 1109.55]
	1-8	1044.91	[1032.93 - 1056.89]
MadMac	1-2	5805.73	[5799.82 - 5811.64]
	1-3	2870.22	[2869.02 - 2871.43]
	1-4	1933.50	[1929.37 - 1937.63]
	1-5	1346.60	[1342.25 - 1350.96]
	1-6	1233.28	[1226.60 - 1239.96]
	1-7	1112.35	[1101.84 - 1122.85]
	1-8	1063.86	[1049.02 - 1078.70]

Table 4: Chain scenario: throughput with different number of hops

4.3.6 The performance anomaly

The last presented scenario presents a fairness issue due to different throughputs on the network (see [3]). In this scenario, two nodes are trying to send their frames at different data rate. The node sending at the lowest rate reduces the throughput of all the nodes transmitting at higher data rate to a value close to the throughput of the slowest node (see Figure 15).

		Throughput (kbps)	Confidence Interval (0.05)
802.11	11Mb/s	1231.74	[1212.54 - 1250.94]
	2Mb/s	1236.13	[1227.64 - 1244.62]
	Total	2467.87	[2453.47 - 2482.27]
MadMac	11Mb/s	1684.09	[1682.90 - 1685.29]
	2Mb/s	842.15	[841.25 - 843.05]
	Total	2526.25	[2524.84 - 2527.65]

Table 5: Performance anomaly: Results

Simulation have been performed with frames of 1000 bytes and two nodes transmitting at 2 Mb/s and 11 Mb/s. Results are given in Table 5. We can see from this table that MadMac provides a better time sharing of the medium and slightly increases the overall throughput. This is due to the fact that the waiting time introduced by MadMac is equal to the time transmission of the packet. Thus, the waiting time for a node transmitting at a low data rate is greater than for the node transmitting at a high data rate. This difference between the waiting times allows a node with smaller waiting time to send more packets. The results obtain with MadMac strongly depends on the *Delta_Slot* value. This point will be discussed in the next section.

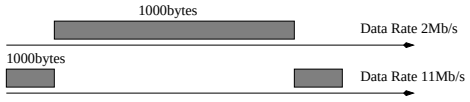

Figure 15: Description of the performance anomaly

5 Discussion

MadMac has many parameters that can be fine tuned to improve its performances. The main problem with fine tuning these parameters is that the modification of one parameter value can improve the protocol performance on only specific scenarios. The first example is the *Delta_Slot* parameter. On one hand, small values of *Delta_Slot* make the protocol

more reactive when some flows stop in the network, but on the other hand, large values of *Delta_Slot* provide a good knowledge of the network. First we have carried out a simulation of 40s with two flows in communication range where one flow stops sending frames at $10s$ and restarts at $20s$. Figure 16 shows the throughput evolution of the other pair that keeps on sending its frames in the network. We can easily see in this figure that the smaller *Delta_Slot*, the more reactive the protocol. This is especially true when the flow stops since the other emitter has to wait for the end of its slot before changing its status and considering that it is alone on the medium. When the flow restarts, then the emission of a single packet changes immediately the status of the second station.

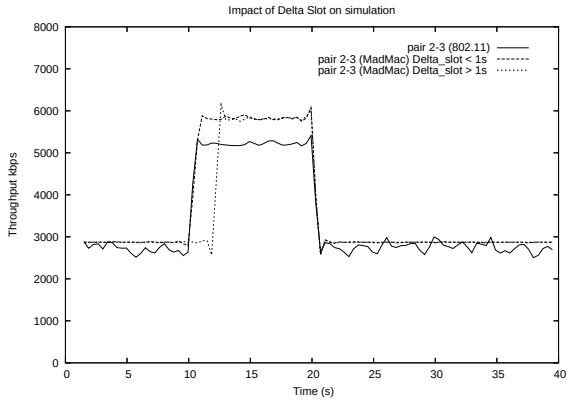

Figure 16: Impact of *Delta_Slot*

To illustrate the conflicting performances that can be obtained with a fine tuning of *Delta_Slot*, we have considered the performance anomaly scenario and the configuration given in Figure 9 with *Delta_Slot* equal to the time transmission of one packet. In the performance anomaly configuration, Table 6 shows the obtained results with this value. The achieved time sharing is better than the one obtained in Table 5 and the overall throughput is increased. Indeed, this small value of *Delta_Slot* allows the node with the high data rate to send more packets in the waiting time introduced by the node with the low data rate. On the other hand, in the scenario of Figure 9, node 1 will experiment collision every *Delta_Slot* because this node sets its $SHARE$ value to 0 and thus does not introduce a waiting time after this *Delta_Slot*.

Other investigations have also been considered with different packet sizes in the network. These different studies show that our protocol still provides a higher global throughput and a better fairness than 802.11 with different packet sizes. Table 7 gives some of the simulation results performed with three pairs within communication range that send packets of different sizes (250, 500 and 1000 bytes). We see that the rates of small packets are increased without sacrificing the overall throughput. Other simulations on this topology but with different numbers of nodes will not be listed here due to space limitations.

		Throughput (kbps)	Confidence Interval (0.05)
MadMac	11Mb/s	2487.33	[2452.40 - 2522.27]
	2Mb/s	655.10	[640.60 - 669.59]
	Total	3142.43	[3117.40 - 3167.46]

Table 6: Performance anomaly results with a small *Delta_Slot*

	size	Throughput (kbps)	Confidence Interval (0.05)
802.11	250	596.60	[586.76 - 606.44]
	500	1194.77	[1177.94 - 1211.59]
	1000	2359.91	[2334.06 - 2385.76]
	Total	4151.29	[4131.15 - 4171.42]
MadMac	250	819.02	[815.71 - 822.34]
	500	1172.05	[1166.99 - 1177.11]
	1000	2401.76	[2393.75 - 2409.76]
	Total	4392.84	[4383.05 - 4402.63]

Table 7: Different packet sizes: results

Investigations about different packet sizes have been extended to the hidden terminal problem. Figure 17 gives the evolution of the throughput of the two hidden pairs with RTS/CTS. This figure shows that MadMac provides a better fairness and a better throughput than 802.11.

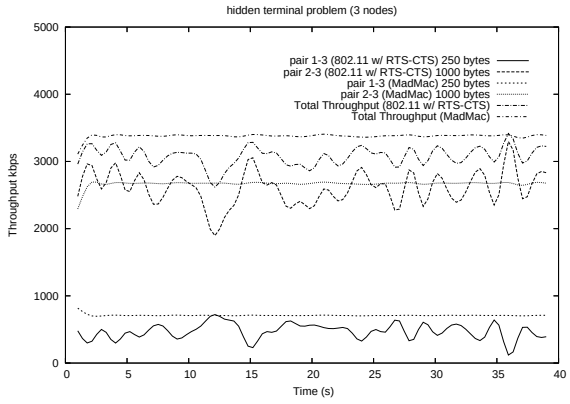

Figure 17: Hidden scenario with different packet size of 250 bytes and 1000 bytes and RTS/CTS enable

Another parameter that can be modified is the triggering of the hidden sending phase depicted in our protocol (see Alg. 1). This phase could be triggered every time an activity is sensed on the medium and not only when the hidden terminal scenario is assumed. This mechanism should be useful to increase fairness and global throughput in the scenario of Figure 9. Indeed, with such a mechanism, the pair $3-4$ will always wait for the pair $1-2$ to send its frame. Conversely triggering this mechanism every time an activity is sensed decreases the throughput in many other configurations.

6 Conclusion

In this paper, we have investigated the trade-off between fairness and capacity in multi-hop ad hoc networks. Since the capacity is not a relevant parameter when considering fairness, we have discussed the notion of fair capacity. We have proposed a new MAC protocol based on 802.11, MadMac, that provides better fairness than 802.11 while maintaining a good aggregate throughput on the network. We have compared MadMac with 802.11 from fairness and throughput points of view. These comparisons have been carried out in many scenarios that are known to lead to fairness issues. Results from these simulations show that in most of the cases MadMac can reach the fair capacity while ensuring fairness among the flows and can achieve a better aggregated throughput than 802.11. The results obtained with MadMac are very promising and many problems, like some famous ones, are solved with its use.

Future works would be to investigate other ad hoc topologies like random topologies and to compare the overall throughput provided by MadMac with the fair capacity. We also plan to compare MadMac to other fair protocols such as PNAV [4] or EHATDMA [15].

Our initial assumptions are very restricting since MadMac considers very limited information (the carrier sensing and the number of collisions). The fairness and the capacity of our protocol can clearly be enhanced with extra information. In the future, we plan to add in MadMac information from other layers of OSI model such as neighbor table from routing layer for instance, in order to measure the impact on the performances.

References

[1] B. Bensaou and Y. Wang and C. C. Ko. Fair medium access in 802.11 based wireless ad-hoc networks. In *MobiHoc*, pages 99–106, Piscataway, NJ, USA, 2000. IEEE Press.

[2] H. Balakrishnan, C.L. Barrett, V.S.A. Kumar, M.V. Marathe, and S. Thite. The distance-2 matching problem and its relationship to the mac-layer capacity of ad hoc wireless networks. *IEEE JSAC*, 22(6):1069–1079, August 2004.

[3] G. Berger-Sabbatel, F. Rousseau, M. Heusse, and A. Duda. Performance anomaly of 802.11b. In *INFOCOM*, 2003.

[4] C. Chaudet and G. Chelius and H. Meunier and D. Simplot-Ryl. Adaptive Probabilistic NAV to Increase Fairness in Ad Hoc 802.11 Mac Layer. In *MedHoc NET*, 2005.

[5] C. Chaudet, D. Dhoutaut, and I. Guérin-Lassous. Performance issues with ieee 802.11 in ad hoc networking. *IEEE Communication Magazine*, to appear, 2005.

[6] D. Qiao and K. Shin. Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE WLAN under the DCF. In *IEEE Int'l Workshop on QoS*, pages pp.227–36., 2002.

[7] Dominique Dhoutaut and Isabelle Guérin Lassous. Impact of Heavy Traffic Beyond Communication Range in Multi-Hops Ad Hoc Networks. In *INC*, Plymouth, Royaume-Uni, July 2002.

[8] F. Cali and M. Conti and E. Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Trans. Netw.*, 8(6):785–799, 2000.

[9] F. Cali and M. Conti and E. Gregori. IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism. *IEEE JSAC*, 18(9), 2000.

[10] Z. Fang and B. Bensaou. Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks. In *INFOCOM*, 2004.

[11] P. Gupta and P. Kumar. Capacity of wireless networks, 1999.

[12] X. L. Huang and B. Bensaou. On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation. In *MobiHoc*, pages 221–231, New York, NY, USA, 2001. ACM Press.

[13] IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems. Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1997.

[14] Information Sciences Institute. NS-2 network simulator. Software Package, 2003. http://www.isi.edu/nsnam/ns/.

[15] H. Jun and H. K. Pung. Fairness properties of medium access control protocols for multi-hop ad hoc wireless networks. *Elsevier publication*, to appear, 2005.

[16] L. Bononi and M. Conti and E. Gregori. Runtime Optimization of IEEE 802.11 Wireless LANs Performance. *IEEE Trans. Parallel Distrib. Syst.*, 15(1):66–80, 2004.

[17] L. Bononi and M. Conti and L. Donatiello. Design and Performance Evaluation of a Distributed Contention Control (DCC) Mechanism for IEEE 802.11 Wireless Local Area Networks. *J. Parallel Distrib. Comput.*, 60(4):407–430, 2000.

[18] J. Li, C. Blake, D. S. J. De Couto, H. Imm Lee, and R. Morris. Capacity of ad hoc wireless networks. In *MobiCom*, pages 61–69, Rome, Italy, July 2001.

[19] Z. Li, S. Nandi, and S. Gupta. Modeling the short-term unfairness of ieee 802.11 in presence of hidden terminals. *Performance Evaluation*, 2005. To appear.

[20] Haiyun Luo, Songwu Lu, and Vaduvur Bharghavan. A new model for packet scheduling in multihop wireless networks. In *MobiCom*, pages 76–86, New York, NY, USA, 2000. ACM Press.

[21] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving mac layer fairness in wireless packet networks. In *MobiCom*, pages 87–98, New York, NY, USA, 2000. ACM Press.

[22] Q. Pang and S. C. Liew and J. Y. B. Lee and V. C. M. Leung. Performance evaluation of an adaptive backoff scheme for WLAN. 4(8):867–879, 2004.

[23] R. Bruno and M. Conti and E. Gregori. A simple protocol for the dynamic tuning of the backoff mechanism in IEEE 802.11 networks: a framework for effective negotiation support in electronic marketplaces. *Comput. Networks*, 37(1):33–44, 2001.

[24] R. Bruno and M. Conti and E. Gregori. Optimal capacity of p-persistent CSMA protocols. *Communications Letters, IEEE*, 7(3):139–141, 2003.

[25] R. Bruno and M. Conti and E. Gregori. Distributed Contention Control in Heterogeneous 802.11b WLANs. In *WONS*, St Moritz, Switzerland, January 2005.

[26] T. Ozugur and M. Naghsineh and P. Kermani and J. A. Copeland. Fair media access for wireless LANs. In *GLOBECOM*, Rio de Janeiro, Brazil, 1999.

[27] V. Bharghavan and A. Demers and S. Shenker and L. Zhang. MACAW: a media access protocol for wireless LAN's. In *SIGCOMM*, pages 212–225, New York, NY, USA, 1994. ACM Press.

[28] Y. Wang and B. Bensaou. Achieving Fairness in IEEE 802.11 DFWMAC with Variable Packet Lengths. In *GLOBECOM*, 2001.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399